Contents

This sample shows how a data source extension is used by DevForce at run time to support multi-tenant scenarios.

- **Platform:** Console
- **Language:** C#, VB
- **Download:** Using DataSourceExtensions (Console)

Problem

Many applications require different operating environments. Multi-tenant applications might maintain a separate environment per tenant. More commonly we see different environments for different phases of deployment e.g., dev, test, stage, and prod.

Solution

DevForce uses the *data source extension* to indicate the specific schema-matching database to use at run time. In this sample, we show how data source extensions can be used to support a multi-tenant application.

Also see the Custom DataSourceKeyResolver sample for an example of how multiple data source extensions can be supported programmatically using the *IDataSourceKeyResolver.*

Running the sample

A simple console application, the app first constructs an EntityManager for the “Tomato” data source name and the “Customer01” data source extension.

```
var entityManager = new DomainModelEntityManager(true, "Customer01");
```

```
Dim entityManager = New DomainModelEntityManager(True, "Customer01")
```

When the query is executed, DevForce will take the key name, “Tomato”, and combine it with the data source extension "Customer01", using an underscore as the separator. So the *DataSourceKey* name that DevForce will look for in the config file will be “Tomato_Customer01”. We’ve provided a connectionString with that name, so DevForce uses that tenant-specific connection information to connect to the database whenever queries or saves are performed against the “Tomato” data source for “Customer01”.

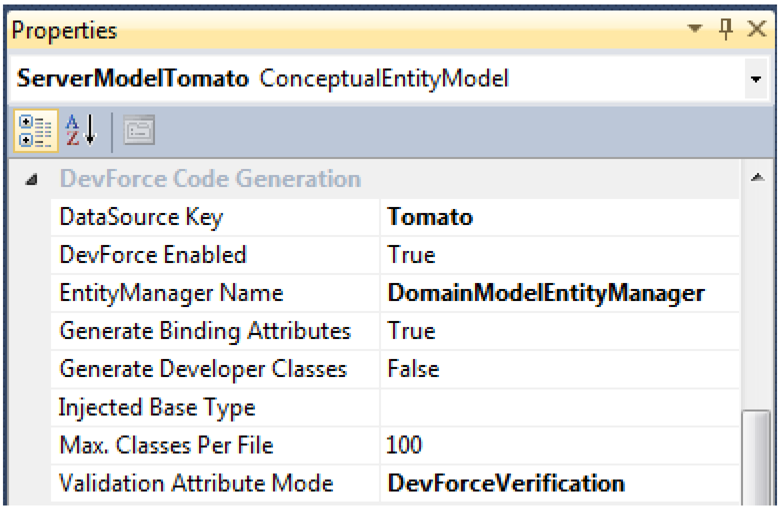
After querying for “Customer01” the sample then goes through the same steps for “Customer02”:

```
var entityManager2 = new DomainModelEntityManager(true, "Customer02");
```

```
Dim entityManager2 = New DomainModelEntityManager(True, "Customer02")
```

Configuration

Let’s look at the DevForce-specific properties on the ServerModelTomato model, specifically noting that the “DataSource Key” is called “Tomato”.



Properties

ServerModelTomato ConceptualEntityModel

DevForce Code Generation

DataSource Key	**Tomato**
DevForce Enabled	True
EntityManager Name	**DomainModelEntityManager**
Generate Binding Attributes	True
Generate Developer Classes	False
Injected Base Type	
Max. Classes Per File	100
Validation Attribute Mode	**DevForceVerification**

DevForce writes this information into the auto-generated model. If you open the *.IB.Designer file you'll see the DataSourceKeyName attribute on generated classes:

```
[IbEm.DataSourceKeyName(@"Tomato")]
public partial class DomainModelEntityManager : IbEm.EntityManager {
```

```
<IbEm.DataSourceKeyName("Tomato")>
Partial Public Class DomainModelEntityManager
Inherits IbEm.EntityManager
```

By default, at run time DevForce will take this key name, and combined with the "data source extension" provided to the EntityManager, try to find a connectionString (or EdmKey) in the config file matching it. So, for example, with a DataSource Key name of "Tomato" and a data source extension of "Customer01", DevForce will look for a connectionString (or EdmKey) named "Tomato_Customer01". (DevForce also performs fall back probing of names when the first choice name is not found; see the Business Object Persistence chapter for more information.)

In the app.config file, we've defined connection strings with the names of the two *DataSourceKeys* we'll use:

```
<connectionStrings>
   <add name="Tomato_Customer01"
       connectionString="…;Initial Catalog=Tomato01…" />
   <add name="Tomato_Customer02"
       connectionString="…;Initial Catalog=Tomato02;…" />
</connectionStrings>
```

We could instead have defined DevForce *EdmKeys* in the ideablade.configuration section of the config file for this information. An *EdmKey* is used by DevForce to represent the runtime connection information to a data source. DevForce will look for *EdmKeys* before looking at *connectionStrings*, so if you've defined the same name in both sections the *EdmKey* will be used and not the connectionString element.

```
<edmKeys>
   <edmKey name="Tomato_Customer01" connection="…;
         Initial Catalog=Tomato01…"/>
</edmKeys>
```

One further point: data source keys are required on the "server side" of a DevForce application. If you're running an n-tier application, the app.config file in your client application does not require connection strings or EdmKeys.

Prerequisites

Attach the two databases in the zip file "TomatoDbInstances.zip" found in the Data folder. These are named "Tomato01" and "Tomato02", and contain a single table named "Tomato". The databases are identical in schema but have some differences in data. They are meant to represent a database for which a separate copy is supplied for each of your customers.